Hand Posture Recognition Using SURF with Adaptive Boosting

Yi Yao
Y.Yao@warwick.ac.uk
Chang-Tsun Li
C-T.Li@warwick.ac.uk

Department of Computer Science,
University of Warwick,
Coventry CV4 7AL, UK

Abstract

An approach making use of SURF feature and Adaboost for hand posture recognition is proposed. First the SURF key points are extracted to describe the blob or ridge-like structures from grey level images. These are potential points of interest that can be used to match with other images with similar structures. Then the statistic parameters of the tendency of gradient changes within small patches surrounding the points of interest are calculated as feature vectors. With all the points of interest, Adaboost is used to train a strong classifier for each posture by selecting the most efficient features, which largely lowers the computational cost of the classification stage. The proposed method was tested on the Triesch Hand Posture Database which is the benchmark in the field. Experimental results showed that our method outperforms existing methods in terms of better recognition accuracy.

1 Introduction

As the need for human-computer interaction (HCI) grows in many aspects of the digital world, various ways of communicating with computers have been exploited and there is little doubt that hand gesture is one of the most intuitive ways for this end. More and more applications in virtual reality (VR), gaming, human-robot interaction (HRI) and HCI based on hand gesture and posture recognition have emerged in the past decade.

In this paper, we focused on hand posture recognition. Many works have been done to improve the performance of hand posture recognition. Viola and Jones [6] successfully developed a face detection framework using Harr-like features and Adaboost to build a strong classifier. Adaboost is used for both feature selection and training of the classifier. This idea has been tested on hand posture recognition by Chieh-Chih Wang and Co-Chih Wang [5] for human robot interaction. They presented a method making use of SIFT [2] to achieve in-plane rotation and scale invariant hand detection. However, their work focused on recognising only 3 hand postures as the commands for communication with robot. The task was not challenging since all 3 postures are very distinct, which makes their method less scalable. Furthermore, their experiments did not test user independency of the method, which is a fairly important criterion of testing hand posture recognition system. Jiatong Bao et al proposed a method for dynamic hand gesture recognition using Speeded-Up Robust Features (SURF) [1] as matching feature to track hand in different frames. Then the hand trajectory used as the main feature to distinguish hand gestures [10]. After extracting

all SURF features from adjacent frames, instead of using the feature selection method, they used all SURF patches in the frame to locate the same hand. The videos with complex background possibly contain areas with sharp edges and dramatic texture changing. Hence applying feature extraction methods based on detecting texture changing extreme points, such as SURF extraction, will produce large number of points of interests from frames of the input video sequence. Hence the procedure of finding a match for every points of interests with 64 dimensional feature vector between two adjacent frames is very time consuming, given that the computational complexity of the procedure is O(M×N), where M and N are number of SURF points in adjacent frames. A method using features based on Modified Census Transform was proposed in 2006 [9]. They trained a simple linear classifier, using feature lookup-tables. But their method did not produce satisfying result on images with complex background. Also, their method did not achieve scale and rotation invariant. Another approach presented by Yikai Fang et al [8]. Their approach is based on the concept of co-training. Since their method is also a boosting based method, the training stage requires large number of samples. They tested their method also on Triesch Hand Posture Database [7], which is the benchmark in this field. But the Triesch Database does not have enough amount of training samples for boosting based methods. Hence they had to enlarge the training set at first by making perturbations on every training sample, which added more computation to the training stage, and made their method less scalable for larger database. Kumar et al [11], proposed a method with novel features which can be extracted by using the computational model of the ventral stream of the visual cortex. Their reported accuracy was 96.35% on uniform background. However the method was only tested on images with uniform background, which is hardly practical for real life scene settings.

The above mentioned methods are those with high reported accuracy on the Triesch Database at one time, and among them the work of Yikai Fang et al [8] and A. Just et al [9] are the ones with highest reported accuracy by now, on different number of training samples. Given the limitations of the afore-reviewed works, we propose the current work with the aims of improving hand posture recognition performance in terms of higher recognition accuracy, lower computational complexity on images with single colored background and also complex backgrounds. The experimental results show that our method outperforms the work of Yikai Fang et al [8] and A. Just et al [9].

2 PROPOSED APPROACH

A method based on SURF and Adaboost is proposed in this paper. First the SURF features are extracted from training samples. Then for each posture, a strong classifier is trained with boosting procedure to find a small combination of features which can distinguish this posture from others most efficiently. Finally the strong classifiers are used to perform classification on test set.

2.1 Feature Extraction

SURF is used in our method as feature. It has in-plane rotation, scale invariant features and certain level of tolerance for view point and illumination changes, which makes it desirable for hand posture recognition. SURF was originally presented by Herbert Bay et al and built upon the idea of SIFT. The interest points are detected at first from the scale

space using Hessian-matrix approximation with the approximate second order Gaussian derivatives which shown in Figure 1 [1].

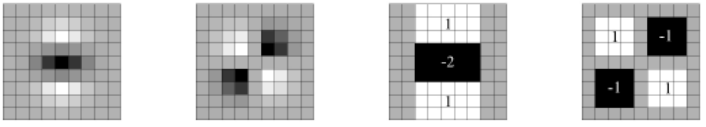

Figure 1 [1]: Approximation of Gaussian second order partial derivative. From left to right: the original second order derivative of Gaussian, y and xy direction; the approximation in y and xy direction, respectively.

SURF uses filters of various sizes to build a scale space, instead of using the same filter iteratively, as shown in Figure 2 [1]. The method called 'integral image' was used for fast convolution. In an integral image, the value of each pixel x is the sum of all pixels from the original image within a rectangular region from the top-left corner pixel to x. With the approximate second order derivatives of the Gaussian filter and the calculated integral image, convolution with Gaussian filters of various sizes can be done at the same speed. Since with the integral image, the summation of intensities inside a rectangular area of any sizes only requires three additions and four accesses to the memory. The candidate points are then picked out of all pixels if they have the extreme value of the determinant of the Hessian matrix in 27 pixels neighbouring area which comprises the upper, lower and current scale levels.

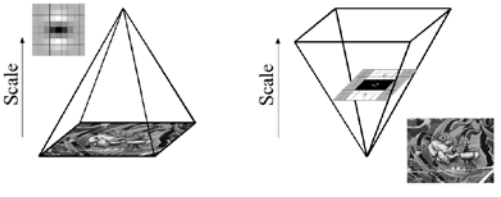

Figure 2 [1]: The difference of scale space structure in SIFT and SURF.

The localisation of interest points is performed next using scale space interpolation. For every interest points, the Haar wavelet response in both x and y direction within a circular neighbourhood with 6s radius of interest points are calculated, where s stands for the scale of the level where this key point is found. Then from a sliding window of size $\pi/3$, the orientation with the largest sum of wavelet response is picked as the dominant orientation. At last for each interest point, Haar wavelet responses within a 20s×20s surrounding region are used to form the final descriptor. The final feature vector of SURF is 64 dimensional, which is half the size of the SIFT descriptor. Figure 3 shows that with similar blob or ridge structures, and despite of the presence of the sleeve or ring on the finger as noise, the same posture from different performers (see the picture on the left-hand side of Figure 3) still have more matched interest points than different postures from the same performer (see the picture on the right-hand side of Figure 3).

There are good reasons why we chose SURF as feature in our method other than SIFT or other features. Firstly, SIFT is rather time consuming. Since SURF uses integral image and LoG approximation, the process of building the scale space is significantly accelerated, even can be paralleled. This makes the computational cost of SURF relatively less sensitive to the image size than SIFT. Secondly, most hand shape features are based on

binary images with hand region enhanced. However, they require segmentation of the hand region from the background. SURF is based on gray scale images, which does not require hand region segmentation, thus largely reducing the computational cost. Thirdly, SURF depends on gradient information on sub-patches, instead of individual gradients, which makes SURF less sensitive to noise. SURF not only uses the sum of wavelet responses, but

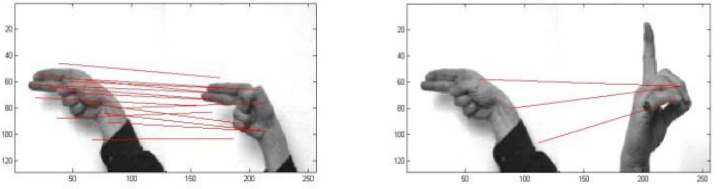

Figure 3: Matched SURF pairs in different postures.

also the sum of absolute values of them in the patches for information of polarity of the intensity changes. As such the descriptor can also indicate the number of changes across the patch, which is shown in Figure 4 [1]. For region on the left, without dramatic intensity changes, the sum of wavelet response and absolute values of them do not have much difference. For region on the right, which has more intensity changes in x direction, the sum of wavelet response will not be changing much, which will fail to show the intensity changing information within the region. On the other hand, the sum of absolute values of wavelet response will be high, which can preserve important information for patch-matching process. Hence for every sub-patches, a 4 dimensional descriptor is built as part of the final feature vector. The most important merit of the proposed method is that it handles the complex background noise without any additional training images with random background [3] or high computational cost of skin color based hand tracking [4]. For those methods using color information, if there are skin colored region in the background, the method will be largely affected. But for our method, the gradient of intensity changes is used other than skin color model, which has much more tolerance on distractions in background. Because in uncontrolled real life scene settings, it is highly possible that people other than posture performer would appearing in the scene. That makes the possibility of regions with exact same texture of target hand shape appearing in the background much less than the possibility of skin-liked region appearing.

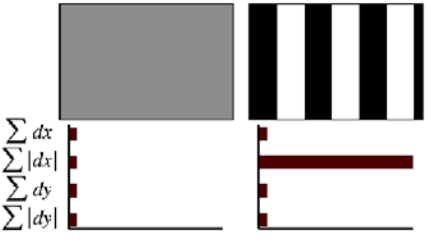

Figure 4 [1]: The sum of absolute values of wavelet response can preserve information of intensity changes in the sub-patches.

In [5], researchers tackled background noise problem by using images with unified background as positive samples. Our method solves this in an even simpler and faster

manner. With Adaboost as the feature selector, we fill the training set with images of complex background. Since all positive and negative training samples have complex background, the selected interest points are the ones that represent blob and ridge-like structures that can best distinguish the target posture from negative samples, with other postures and background textures. Hence the selected patches are the ones appear in positive samples with high frequency and almost never appear in other postures and random background. Experimental results showed that this method can achieve high accuracy.

2.2 Classifier training

Regardless of the relatively small 64 dimensional feature vector of SURF, even images of size as small as 128×128 pixels still dozens of interest points can be detected as features. Without feature selection process to sift out the useful ones, the computational cost is enormous. Boosting based methods can be used to find a subset of all points of interests, which is tested to be most effective. Adaboost is one of the most popular boosting learning algorithms, and based upon the idea of Adaboost we build a training process for classifier.

The feature selection mechanism of proposed method is described in details in Algorithm 1. Assuming there are X images in the training set, and Y postures in total, and let T be the maximum number of weak classifiers for one posture. Firstly the SURF features are extracted from all positive samples in the training set. Different weights will be assigned to all samples in training set. All positive samples share the same initial weight, and so are the negative samples. For every target posture, the SURF vectors of all positive samples labelled with this posture will be put into the weak classifier pool. All SURF vectors which are the weak classifiers of this target posture in the pool will be tested to label all samples in the training set. To evaluate the performance of each SURF vectors, a error rate will be calculated for every vector using the weight of all training samples. One SURF vector with the lowest error rate will be chosen as a weak classifier $h_t(I)$ of the strong classifier of this posture. Then the weight of all training samples will be updated based on the error rate of the chosen SURF vector. The weights of those samples were correctly classified by the chosen vector will be reduced by a factor. For those were not correctly classified, the weight will stay large. The process iterates until the error rate of latest chosen vector is smaller than a threshold or the number of selected vector reaches a certain limits. In every iteration, since the wrongly classified samples have large weights, then the process will looking for next weak classifier which can specifically classify those samples with large weight. This process has fairly intuitive pattern. Every chosen SURF vector is specifically suitable for classifying samples with certain characteristics, which cannot be efficiently classified by other chosen vectors. Therefore for every target postures the selected weak classifier will form a strong classifier. Every weak classifier consists of a weight, a selected SURF feature and its threshold.

In the process of selecting weak classifiers, for every chosen vector, a optimised threshold will be evaluated. The threshold is the Euclidean distance between the selected vector and the first matched vector on the matching score list, divided by the distance of selected vector to the second best matched vector on the list. This threshold represents how large the difference is between the matching score of best matched interest point and that of second to best interest point. This threshold indicates the uniqueness of this selected vector and the efficiency of classification using this vector. The values threshold between 0.20 to 0.95 are tested for every selected SURF vector. After this SURF vector is selected, a weight will be assigned for classification stage later. This weight defines how important

this weak classifier is. The larger the error rate is, the smaller the weight will be, which indicates the importance of this weak classifier is lower.

Training process for all target postures

Input: The training set consists of X images and their corresponding posture sets: (I_1, P_1), ... , (I_X, P_X), where I_x is the xth image and P_x is the posture label set of I_x , $P_x = \{p_{x,y} | y = 1,2, ... Y\}$. If posture y appearing in the xth image, $p_{x,y}$ = 1, otherwise $p_{x,y}$ = 0.

1. **for** y = 1, ... ,Y **do**
2. Initialise weights $w_{1,x} = 1/2N_p$ for positive samples, $w_{1,x} = 1/2N_n$ for negative samples, where N_p and N_n are number of positives and negatives respectively;
3. **for** t = 1, ... , T **do**
4. Initialise the error rate e_t = 1;
5. **while** e_t < = Error Rate Threshold
6. Normalise weight of all training samples, so that

$$\sum_{x=1}^{X} w_{t,x} = 1;$$

7. Select one feature (f_t) and its threshold (θ_t), from the SURF interest points of all positive samples, which minimise the error rate:

$$e_t = \sum_{x=1}^{X} w_{t,x} \left| h_t(I_x, f_t, \theta_t) - p_{x,y} \right| \quad (1)$$

8. For each weak classifier $h_t(I)$, where f_t, θ_t are minimisers of e_t, assign weight α_t:

$$\alpha_t = \log \frac{1-e_t}{e_t} \quad (2)$$

Put $h_t(I) = h(I, f_t, \theta_t, \alpha_t)$ into $H_y(I)$, which is the selected feature set (strong classifier) of the yth posture;

9. Update the weights of all training samples:

$$w_{t+1,x} = w_{t,x} \left(\frac{e_t}{1-e_t} \right)^{1-\lambda} \quad (3)$$

Where λ= 0 if sample I_i is correctly classified, λ = 1, otherwise;

10. **End while**
11. **End for**
12. **End for**
13. **return** strong classifier $H(I) = (H_1(I), ... , H_Y(I))$.

Algorithm 1. The training process of strong classifiers for all target postures.

The value of *T* in the training stage, needs to be evaluated in the experiments. In order to guarantee the efficiency of the classification stage, we need to terminate the training process at some point when the number of selected weak classifiers is larger than the threshold *T*, but the error rate is still not low enough. *T* is set to 20 in our experiment. That means this training process only picks necessary amount of weak classifiers. The amount of selected weak classifiers for every target posture could be different.

2.3 Classification

The proposed method incurs even less computation in the classification stage than [5] due to the various number of weak classifiers *h_t (I)* involved in every strong classifier *H_y (I)* of each posture in the training process. This is because the training process terminates when the error rate *e_t* is lower than the predefined threshold, every *H_y (I)* can achieve the target accuracy without wasting time on selecting a certain number of weak classifiers for every posture. That means in classification stage the computation will focused on those posture with relatively large number of weak classifiers, which could be very important for pattern recognition tasks require real time response rate like hand posture and gesture recognition. For postures with higher similarity and relatively harder to be classified, like posture 'I' and 'Y' in Triesch database, there will be more features picked out for the classification task. Shown in Figure 2, posture 'I' has 19 selected features. For posture 'L', only 4 features selected. Because the SURF features of posture 'L' are so unique that only using 4 of them can distinguish this posture from the others quite well.

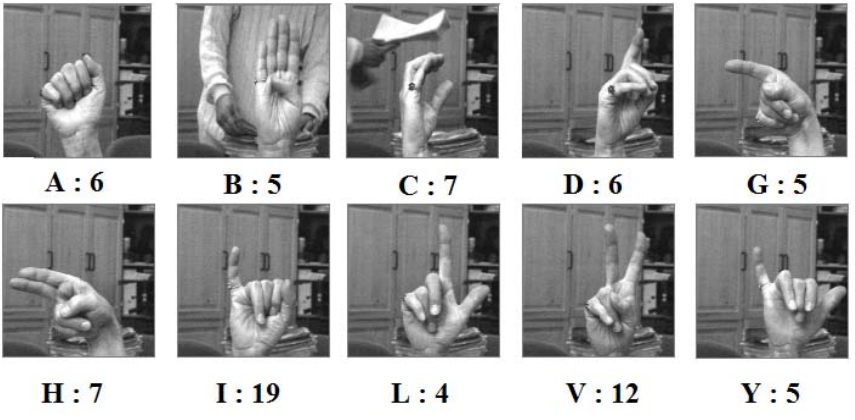

Figure 5. Number of weak classifiers within trained strong classifier for all 10 postures with complex background in Triesch Hand Posture Database.

The classification mechanism is shown with details in Algorithm 2. Given a testing image, firstly the SURF feature vectors will be calculated. Then for each strong classifier, all weak classifiers within it will produce a matching result, which is 1 or 0, according to the thresholds of all weak classifiers. This score will have the weight of this weak classifier. Then the weighted sum of all weak classifiers in this strong classifier, i.e.,

$$H_y(I) = \sum_{t=1}^{T} \alpha_t \times h_t \qquad (4)$$

will be produced as matching result of this strong classifier. If *H_y (I)* is larger than a certain percentage μ of sum of weights

$$\mu \sum_{t=1}^{T} \alpha_t \qquad (5)$$

then output of *H_y (I)* is 1. The parameter μ needs to be estimated through experiments. If a learning algorithm like EM is used here to estimate the optimal value of μ, the computational cost of the proposed method can be shortened even further. If there are more than one classifier producing 1 as result, the actual weighted sum in Eq.(4) from these classifiers will be compared and the one with the highest sum wins.

Classification with SURF

Input: A testing image: I, with the strong classifier $H(I) = (H_1(I), \ldots, H_Y(I))$ from the training stage, where $H_y(I)$, y = 1,2, ... Y, consists of T_y weak classifiers $h_t(I) = h(I, f_t, \theta_t, \alpha_t)$, α_t, f_t, θ_t indicating the weight, selected SURF interest point and threshold respectively.

1. Extract the SURF features set $S = s_1, \ldots s_M$, from I;
2. Initialise WeightSum = 0;
3. **for** y = 1, ... , Y **do**
4. **for** t = 1, ... , T_y **do**
5. Find the nearest match of h_t, s_m in S;
   if Euclidean Distance d(h_t, s_m) < θ_t, **then**
   WeightSum := WeightSum + α_t;
6. **End if**
7. **End for**
8. **End for**
9. **if** WeightSum > $\mu \sum_{t=1}^{T} \alpha_t$ **then**
10. **return** 1
11. **else**
12. **return** 0
13. **end if**

Algorithm 2. The classification stage with strong classifiers of all target postures trained by Algorithm 1.

3 EXPERIMENTAL RESULTS

Our method is independent of scale, in-plane rotation and hand position in the scene. But we still choose to test this method on Triesch Hand Posture Database [7] as it is the benchmark of hand posture recognition, although all postures are up-right in the scene. The test of using SIFT on hand posture recognition has been done in [5], but only 3 target postures were used, and user independency was not well tested. So we decided to test the proposed method on a larger number of performers in the Triesch database. This database

consists of 10 hand postures performed by 24 persons. There are three kinds of background settings, namely uniform light, uniform dark and complex. In total, there are 720 greyscale images of 128×128 pixels. Samples of the database are shown in Figure 5. There are 72 images for each posture. We test the proposed method in two different experiments.

Experiment 1: For comparison with work of Yikai Fang et al [8], we apply the exact same experiment setting as in [8]. For each posture, the training set consists all 30 images from 10 performers and the remaining images of the other 14 performers form the testing set. Also for the convenience of comparison with A. Just et al [9], the same experimental setting is deployed in Experiment 2. For each posture the training set has images of 8 performers and the testing set has 16 performers' image. The results are shown in Table 1 and 2, respectively.

	Light	Dark	Complex	Average
Our method	93.9%	94.4%	90.2%	92.8%
Reported in [8]	-	-	-	90.1%

Table 1. Results of Experiment 1

	Light	Dark	Complex	Average
Our method	93.6%	94.3%	90.0%	92.6%
Reported in [9]	92.8%	92.8%	81.3%	89.0%

Table 2. Results of Experiment 2

In Experiment 1, the proposed method outperforms [8] (They only reported the average accuracy which is 90.1%) with the same number of training and testing images. Furthermore, better than method in [8], our method does not require any perturbations work on training images. That makes the proposed method more efficient and scalable. Producing decent accuracy with small training set is fairly rare for boosting based methods, which indicates out method can be further generalised on dynamic hand posture and dynamic hand gesture recognition, which are based on video sequences. The boosting method with SURF could serve well, since the training methods requiring large number of video sequences are not practical.

With the same experimental setting with [9] in Experiment 2, the proposed method outperformed [9]. Our experimental results revealed that the proposed method can achieve higher accuracy without a large number of training samples or perturbation on the training set. Especially on complex background images, our method produced 92.6% of truth positive.

4 CONCLUSION

In this paper, SURF is used in conjunction with Adaptive Boosting for recognising hand postures in greyscale images with various backgrounds. Matching the tendency of gradient changes within a small neighbourhood of interest points has been successfully applied to tackle the varying complex background. The segmentation stage of hand region is spared

and only the most efficient features from the feature pool are selected to build the strong classifier. The proposed method achieved high recognition accuracy on the Triesch Hand Posture Database at a low computational cost with a small training set, without including any additional training images of complex backgrounds or perturbation on the training set.

References

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, SURF: Speeded-Up Robust Features, *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346-359, 2008.

[2] David G. Lowe, Distinctive image features from scale - invariant keypoints, *International Journal of Computer Vision*, 60(2), pp. 91-110, 2004.

[3] L. Anton-Canalis and E. Sanchez-Nielsen, Hand posture dataset creation for gesture recognition, *International Conference on Computer Vision Theory and Applications (VISAPP'06)*, Set´ubal, Portugal, February 2006.

[4] Mahmoud Elmezain, Ayoub Al-Hamadi, Bernd Michaelis, Hand trajectory-based gesture spotting and recognition using HMM, *16th IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, 2009.

[5] Chieh-Chih Wang, Co-Chih Wang, Hand Posture Recognition Using Adaboost with SIFT for Human Robot Interaction, *Proceedings of the International Conference on Advanced Robotics (ICAR'07)*, Jeju, Korea, 2007.

[6] P. Viola and M. J. Jones, Robust real-time face detection, *International Journal of Computer Vision*, vol. 57(2), pp. 137-154, 2004.

[7] J. Triesch and C. V. D. Malsburg, Robust classification of hand postures against complex backgrounds, *Second International Conference on Automatic Face and Gesture Recognition*, pp. 170–175, 1996.

[8] Yikai Fang, et al., Hand posture recognition with co-training, *19th International Conference on Pattern Recognition*, Tampa, FL, 2008.

[9] A. Just, Y. Rodriguez, and S. Marcel, Hand posture classification and recognition using the modified census transform, *7th International Conference on Automatic Face and Gesture Recognition*, pp. 351–356, 2006.

[10] Jiatong Bao, et al., Dynamic hand gesture recognition based on SURF tracking, *2011 International Conference on Electric Information and Control Engineering (ICEICE)*, pp.338–341, 2011.

[11] Pramod P. Kumar, Prahlad Vadakkepat, Loh A. Poh, Graph matching based hand posture recognition using neuro-biologically inspired features, *2010 11th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 1151–1156, 2010.